ECE 171A: Linear Control System Theory
Lecture 2: ODEs and Cruise Control

Yang Zheng

Assistant Professor, ECE, UCSD

April 03, 2024

Reading materials: Ch 1.6, Ch 4.1, Ch 6.2

Announcements

- Office hours:
  - 6:30 pm - 8:30 pm, Tuesdays at FAH 3002; Yang Zheng
  - 6:30 pm - 8:30 pm, Thursdays at FAH 3002; Chih-Fan (Rich) Pai

Ideally, I would like most of you, if not all, to go to the office hours together even if you don't have questions. You can even help us answer questions from others.

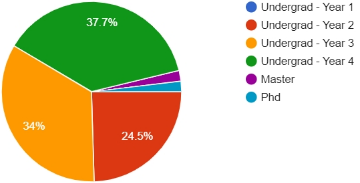
It is important to have a supportive community for this class!

- Discussion session at 1:00 pm this afternoon: Review on ODEs (I)

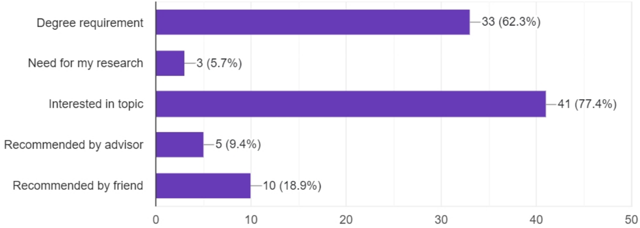
Outline

Background survey

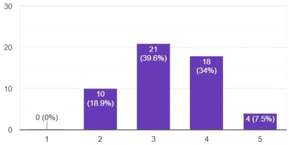
- Years



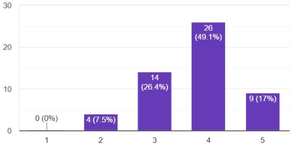
- Reasons for taking this class



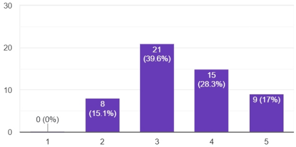
Background survey



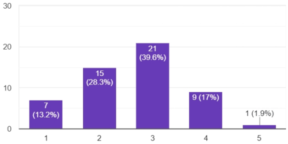
(a) Signals and Systems



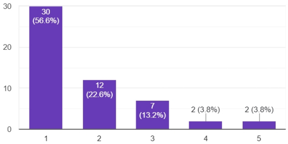
(b) Coding experience



(c) Linear algebra



(d) ODEs



(e) Convex Optimization (not required in this class)

Background survey

- Are there any specific applications of feedback and control concepts that you are interested in?
  - Autonomous vehicles, robotics, and humanoid robots.
  - automatic center of mass adjustment robot
  - Electric vehicles
  - Machine and reinforcement learning
  - ......
- Suggestions we have got so far
  - Some coding example during lecture or DI.
  - Responsive on Piazza
  - Putting student wellbeing first is very much appreciated.
  - examples in class that can help with better understanding concepts/coursework
  - Engaging and available outside the class
  - In-depth coverage of bode plots and related topics
  - ......

Thank you for the feedback!

Outline

Ordinary Differential Equations

As we will see, the behavior of many physical systems can be described using ordinary differential equations in the time domain

- In the frequency domain, they become **transfer functions** (Week 4)

- A **differential equation** is any equation involving a function and its derivatives, e.g.,
$$\frac{d}{dt}x(t) = -x(t). \tag{1}$$
- A **solution to a differential equation** is any function that satisfies the equation, e.g., for (1), we have
  – A solution is
  $$x(t) = e^{-t}$$
  – Another solution is
  $$x(t) = 2e^{-t}$$
  – A general solution is
  $$x(t) = e^{-t}x(0),$$
  where $x(0) \in \mathbb{R}$ is the initial value at $t = 0$.

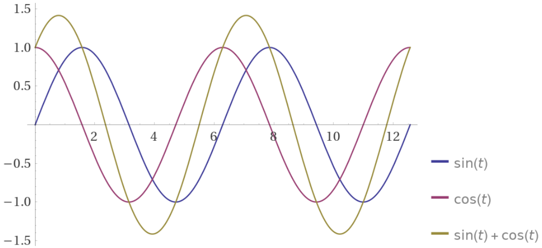
An second-order example

Example

- Consider a linear ODE: $\frac{d^2}{dt^2}x(t) + x(t) = 0$
- Two particular solutions are:

$$x_1(t) = \cos(t) \quad \Rightarrow \quad \frac{d^2}{dt^2}\cos(t) = -\cos(t),$$
$$x_2(t) = \sin(t) \quad \Rightarrow \quad \frac{d^2}{dt^2}\sin(t) = -\sin(t).$$

- In fact, $x(t) = c_1 x_1(t) + c_2 x_2(t)$ with $c_1, c_2 \in \mathbb{R}$ is also a solution



Ordinary Differential Equations

- An n**th-order linear ordinary differential equation (ODE)** is:

$$\frac{d^n}{dt^n}y(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}y(t) + \ldots + a_1\frac{d}{dt}y(t) + a_0 y(t) = u(t) \qquad (2)$$

- If $u(t) = 0$, then the nth-order linear ODE is called **homogeneous**
- A **particular solution** is a solution $y(t)$ that contains no arbitrary constants
- A **general solution** is a solution $y(t)$ that contains n arbitrary constants

Definition (Initial value problem)

An ODE (2) together with initial value constraints

$$y(t_0) = y_0, \quad \dot{y}(t_0) = y_1, \quad \ldots \quad y^{(n-1)}(t_0) = y_{n-1}.$$

Theorem

Let $u(t)$ be a continuous function on an interval $\mathcal{I} = [t_1, t_2]$. Then, for any $t_0 \in \mathcal{I}$, a solution $y(t)$ of the initial value problem exists on $\mathcal{I}$ and is unique.

State-space model

Any n**th-order linear ODE** can be reformulated into a first-order matrix ODE of the form

$$\dot{x}(t) = Ax(t) + Bu(t)$$

- Define variables:

$$x_1(t) = y(t), \qquad x_2(t) = \frac{d}{dt}y(t), \qquad \dots \qquad x_n(t) = \frac{d^{n-1}}{dt^{n-1}}y(t)$$

- The linear ODE (2) specifies the following relationships:

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= x_3(t) \\
&\vdots \\
\dot{x}_{n-1}(t) &= x_n(t) \\
\dot{x}_n(t) &= -a_0x_1(t) - a_1x_2(t) - \cdots - a_{n-1}x_n(t) + u(t)
\end{aligned}$$

State Space Model

- Let $x(t) := \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_n(t) \end{bmatrix}^\mathsf{T}$ be a vector called **system state**
- A **state space model** of the linear ODE is obtained by re-writing the equations in vector-matrix form:

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{B} u(t)$$

- **Matrix exponential:**

$$e^A := I + A + \frac{1}{2}A^2 + \ldots + \frac{1}{n!}A^n + \ldots \qquad \Rightarrow \qquad \frac{d}{dt}e^{At} = Ae^{At}.$$

- A general solution to $\dot{x}(t) = Ax(t)$ is $x(t) = e^{At}x(0)$ (Week 3/4).

More analysis of state-space models will be discussed in ECE 171B

$$\dot{x}(t) = Ax(t) + Bu(t)$$

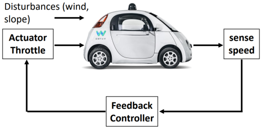
Outline

Cruise Control

Cruise control is a common feedback system encountered in everyday life,

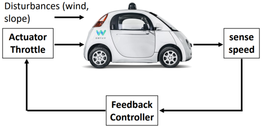
- It aims to maintain a constant velocity in the presence of disturbances caused by road slop/frictions/air drag etc.

Control goals



- **Stability/performance**
  - Steady state velocity approaches desired velocity
  - Smooth response; no overshoot or oscillations
- **Disturbance rejection**
  - Effect of disturbances (e.g., hills) approaches zero
- **Robustness**
  - Results don't depend on the specific values of the system parameters

Cruise control - Modeling



Parameters, input/output variables (simplified)

- Desired speed: v_{des}
- System variable (output): speed v
- System parameter: mass m (which may change)
- Disturbance: road slop $F_{\text{hill}} = -mg\sin(\theta)$, air drag $F = -\delta \times v$
- Actuator (input): Engine/Braking Force F_{engine}

System model

$$m\dot{v} = F_{\text{engine}} - \delta \times v - mg\sin(\theta)$$

or we can also include the vehicle's position explicitly

$$\dot{p} = v,$$
$$\dot{v} = -\frac{\delta}{m}v - g\sin(\theta) + \frac{1}{m}F_{\text{engine}},$$

Experiments:

- $v_0 = 10m/s$, $m = 500\text{kg}, \delta = 0.5, \theta = 0$
- Goal $v_{\text{des}} = 15m/s$; How to design F_{engine}?

Cruise control - Simulation

Simulate the dynamics

- Matlab ODE45 function: `[t,y] = ode45(odefun,tspan,y0)`

Strategies

- Feedforward (open-loop) control

$$F_{\text{engine}} = \begin{cases} 800 & \text{if} \quad 0 \leq t \leq 5s \\ 0 & \text{otherwise} \end{cases}$$

- Feedback (closed-loop) control: based on deviation $e(t) = v_{\text{des}} - v(t)$
  - **P (Proportional)** control, e.g.,

$$F_{\text{engine}} = K_{\text{p}} e(t)$$

  - **I (Integral)** control

$$F_{\text{engine}} = K_{\text{i}} \int_0^t e(t) dt$$

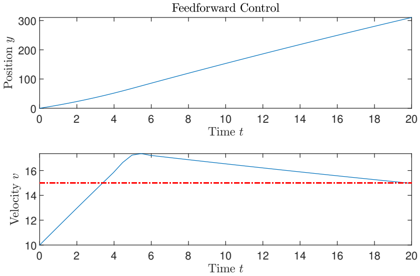
  - **D (Derivative)** control

$$F_{\text{engine}} = K_{\text{d}} \frac{d}{dt} e(t)$$

Cruise control - Feedforward control

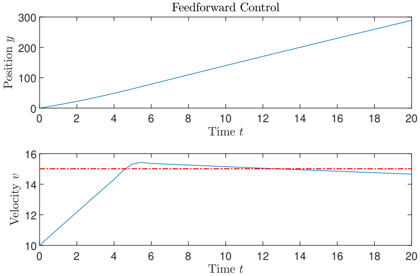
Case 1:

$$F_{\text{engine}} = \begin{cases} 800 & \text{if} \quad 0 \leq t \leq 5s \\ 0 & \text{otherwise} \end{cases}$$



Case 2:

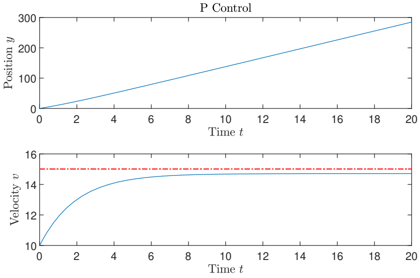
$$F_{\text{engine}} = \begin{cases} 600 & \text{if} \quad 0 \leq t \leq 5s \\ 50 & \text{otherwise} \end{cases}$$



Cruise control - P control

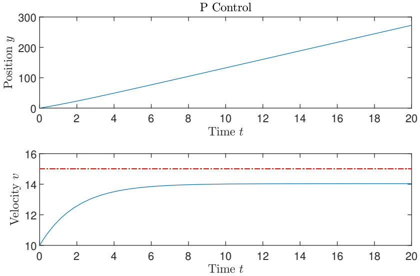
Case 1: flat road $\theta = 0$

$$F_{\mathrm{engine}} = K_{\mathrm{p}} e(t)$$



Case 2: uphill $\theta = 2^\circ$

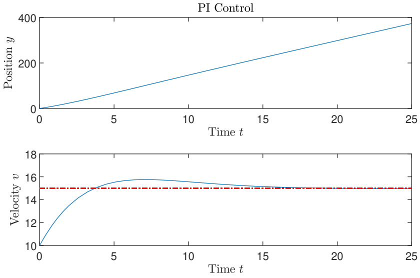
$$F_{\mathrm{engine}} = K_{\mathrm{p}} e(t)$$



Cruise control - PI control

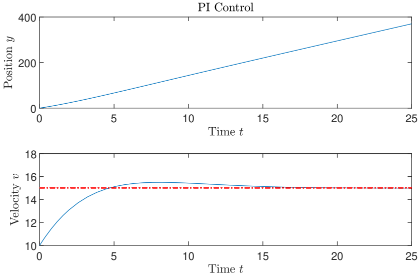
Case 1: flat road $\theta = 0$

$$F_{\text{engine}} = K_{\text{p}} e(t) + K_{\text{i}} \int_0^t e(t) dt$$



Case 2: uphill $\theta = 2^\circ$

$$F_{\text{engine}} = K_{\text{p}} e(t) + K_{\text{i}} \int_0^t e(t) dt$$



PID controller

- **P controller**: faster response (larger control gain) $\rightarrow$ more oscillated transient behavior; but fail to reach the desired value;
  - A proportional controller will reduce the rise time but cannot eliminate the steady-state error.
- **I controller**: reach the desired steady state; faster response (larger control gain) $\rightarrow$ overshoot and oscillation;
  - An integral controllers tend to respond slowly at first, but over a long period of time they tend to eliminate errors.
  - The integral controller eliminates the steady-state error, but may make the transient response worse.
- **D controller**: improve the transient dynamics (no experiments today)
  - A derivative controller will in general have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.
  - The derivative controller is never used alone. With sudden changes in the system, the derivative controller will compensate the output fast.

More analysis on PID will be discussed later.

Outline

Summary

Review on ODEs

- An n**th-order linear ordinary differential equation (ODE)** is:

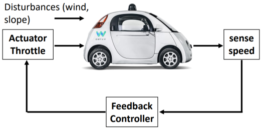
$$\frac{d^n}{dt^n}y(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}y(t) + \ldots + a_1\frac{d}{dt}y(t) + a_0 y(t) = u(t)$$

- First-order matrix ODE

$$\dot{x} = Ax(t) + Bu(t)$$

Cruise control

P control $F_{\text{engine}} = K_{\text{p}}e(t)$
I control $F_{\text{engine}} = K_{\text{i}}\int_0^t e(t)dt$
D control $F_{\text{engine}} = K_{\text{d}}\frac{d}{dt}e(t)$



Feedback control = Sensing + Computation + Actuation